```
! Copyright (C) 2010 John Benediktsson
! See http://factorcode.org/license.txt for BSD license

USING: assocs calendar combinators environment formatting
grouping io io.files kernel make math math.parser math.ranges
sequences splitting xml.entities ;

IN: text-to-pdf

<PRIVATE

: pdf-string ( str -- str' )
    H{
        { HEX: 08    "\\b"  }
        { HEX: 0c    "\\f"  }
        { CHAR: \n   "\\n"  }
        { CHAR: \r   "\\r"  }
        { CHAR: \t   "\\t"  }
        { CHAR: \\   "\\\\" }
        { CHAR: (    "\\("  }
        { CHAR: )    "\\)"  }
    } escape-string-by "(" ")" surround ;

: pdf-object ( str n -- str' )
    "%d 0 obj\n" sprintf "\nendobj" surround ;

: pdf-stream ( str -- str' )
    [ length 1 + "<<\n/Length %d\n>>" sprintf ]
    [ "\nstream\n" "\nendstream" surround ] bi append ; inline

: pdf-info ( -- str )
    [
        "<<" ,
        "/CreationDate D:" now "%Y%m%d%H%M%S" strftime append ,
        "/Producer (Factor)" ,
        "/Author " "USER" os-env "unknown" or pdf-string append ,
        "/Creator (created with Factor)" ,
        ">>" ,
    ] { } make "\n" join ;

: pdf-catalog ( -- str )
    {
        "<<"
        "/Type /Catalog"
        "/Pages 4 0 R"
        ">>"
    } "\n" join ;

: pdf-font ( -- str )
    {
        "<<"
        "/Type /Font"
        "/Subtype /Type1"
        "/BaseFont /Courier"
        ">>"
    } "\n" join ;
```

```
: pdf-pages ( n -- str )
    [
        "<<" ,
        "/Type /Pages" ,
        "/MediaBox [ 0 0 612 792 ]" ,
        [ "/Count %d" sprintf , ]
        [
            5 swap 2 range boa
            [ "%d 0 R " sprintf ] map concat
            "/Kids [ " "]" surround ,
        ] bi
        ">>" ,
    ] { } make "\n" join ;

: pdf-text ( lines -- str )
    [
        "BT" ,
        "54 738 Td" ,
        "/F1 10 Tf" ,
        "12 TL" ,
        [ pdf-string "'" append , ] each
        "ET" ,
    ] { } make "\n" join pdf-stream ;

: pdf-page ( n -- page )
    [
        "<<" ,
        "/Type /Page" ,
        "/Parent 4 0 R" ,
        1 + "/Contents %d 0 R" sprintf ,
        "/Resources << /Font << /F1 3 0 R >> >>" ,
        ">>" ,
    ] { } make "\n" join ;

: pdf-trailer ( objects -- )
    [
        "xref" ,
        dup length 1 + "0 %d" sprintf ,
        "0000000000 65535 f\r" , ! first
        9 over [
            over >hex "%010s 00000 n" sprintf , length 1 + +
        ] each drop
        "trailer" ,
        "<<" ,
        dup length 1 + "/Size %d" sprintf ,
        "/Info 1 0 R" ,
        "/Root 2 0 R" ,
        ">>" ,
        "startxref" ,
        [ length 1 + ] map-sum 9 + "%d" sprintf ,
        "%%EOF" ,
    ] { } make "\n" join print ;

: string>lines ( str -- lines )
    "\t" split "    " join string-lines
    [ [ " " ] when-empty ] map ;
```

```
: lines>pages ( lines -- pages )
    [ 84 <groups> ] map concat 57 <groups> ;

: pages>objects ( pages -- objects )
    [
        pdf-info ,
        pdf-catalog ,
        pdf-font ,
        dup length pdf-pages ,
        dup length 5 swap 2 range boa zip
        [ pdf-page , pdf-text , ] assoc-each
    ] { } make
    dup length [1,b] zip [ first2 pdf-object ] map ;

: write-pdf ( objects -- )
    [ "%PDF-1.4" print [ print ] each ] [ pdf-trailer ] bi ;

PRIVATE>

: text-to-pdf ( str -- )
    string>lines lines>pages pages>objects write-pdf ;

: file-to-pdf ( path encoding -- )
    file-contents text-to-pdf ;
```